Method and System for Data Management

TECHNICAL FIELD OF THE INVENTION

The invention relates generally to the use of databases. The invention relates preferably to a method and a system associated with the use of databases with mobile terminals.

BACKGROUND ART OF THE INVENTION

The following notions are used in this application:

"Data management system" is an entity, which comprises one or more databases and/or data management systems, whereby the system is responsible for reading the data structures contained in the databases and/or data management systems and for changing these data structures.

"Database" is an information structure, which comprises one or more data elements, and the use of which is controlled by the data management system. The invention is applicable both in relational databases and in databases of other forms, such as in object oriented databases.

"Data element" is an information structure, which can comprise other data elements or such data elements, which can be construed as atomary data elements. For instance, in a relational database data elements are represented by tables comprising rows. The rows comprise fields, which are typically atomary data elements.

"Database operation" is an event, during which data elements are read from the database, during which data elements of the database are modified, during which data elements are removed from the database, and/or during which data elements are added to the database.

"Transaction" is a plurality of database operations acting on the data elements. A transaction can also comprise further transactions.

"Database Catalog" is a logical database within a database instance. A physical database can manage data of multiple database catalogs. Each database catalog can act as an independent master or replica database node in a database synchronization environment.

"Database Schema" is the structure of a database system, described in a formal language supported by the database management system (DBMS). In a relational database, the schema defines the tables, the fields in each table, and the relationships between fields and tables.

"Master database" is a database catalog in a database synchronization system that contains the official version of synchronized/distributed data. A master database can have multiple replica databases.

"Replica database" is a database catalog in a database synchronization system that contains a full or partial tentative copy of the master data.

"Synchronization" is operation between replica and master database catalogs in which changed data is exchanged between the catalogs. In one known embodiment, this means propagation of Intelligent Transactions from replica to master and/or subscription of changed data of publications from master to replica.

"Publication" is a set of data in a database catalog that has been published in master database for synchronization to one or multiple replica databases.

There are presently some software push technologies available that are capable of pushing a set of software files to terminals and keeping the once-installed configuration up-to-date by pushing upgrade files to the terminal on an as-needed basis. This means that the software that is stored and run locally in the terminal can be managed remotely.

However, this technology does not address the requirements of mobile users and wireless communities where it is important to manage a potentially large, dynamically changing set of applications. For instance, the list of services available to that terminal, i.e. the "desktop" of the terminal, is relatively static and cannot be easily changed to reflect different roles and locations of the user of the terminal.

Many web portals allow personalization of their home pages so that different users can have different view to the list of services provided by the portal. This perzonalization data is separate for each portal and it cannot be utilized when using another portal. Moreover, web portal personalization techniques are applicable only to web-based applications, i.e. services that are provided by a server that resides in the network.

For these reasons, the current web service personalization technology that operates on isolated and proprietary user profile data is not feasible in environments, where there are potentially a very large number of application service providers, e.g. one per each wireless base station, which all need to conform with the personal preferences of the user and capabilities of the current terminal of the user.

SUMMARY OF THE INVENTION

The objective of this invention is to present a method and a system which allows managing a large dynamically changeable set of applications in mobile Internet.

The objectives of the invention are attained by specifying user preferences with an identity server, and matching the preferences with applications of a community server. The result of the match is stored in a service assembly point. This invention also introduces the concept of Identity Data, which is maintained in the Identity Server, and is an essential part in process of joining to a new community. Examples of Identity Data may include name, address, age, size, weight, sex, profession, hobbies, personal interests, etc.

The invention provides a service matchmaking method that efficiently matches the users preferences and available services of a community into a list of local and web-based services that is of interest to the user.

The invention thus provides a solution to the problem of managing services and configuration of smart network node in environments where the services and their content data need to be managed remotely from multiple remote sources in a dynamic manner based on preferences of the user and capabilities of the currently used terminal. The related service data may include the availability, registered users, application binary files, configuration, parameter requirements, classification, etc.

A typical example of a dynamically configurable smart network node according to the invention is a so-called smart phone. The user of a smart phone has different roles in different communities depending on location, time of day and personal preferences. For example, during workdays from 8AM to 5PM, a person can have a "Boss" role in a "Work" community. In this role, the person wants to have access to corporate intraweb, e-mail and chat applications through his terminal. Outside business hours, this same person may want to use applications that are available to him via the "Dad" role in his "Home" community. The services may run in the network servers or alternatively, they can run locally in the smart terminals. To ensure ease of use of the terminal, managing the "desktop", i.e. determining the services available to the user in his current role, should be done centrally in each of the communities. No or minimal amount of user interaction is necessary to manage the configuration of the terminal.

With the present invention it is possible to remotely manage software configuration and content data of a terminal from various community servers and keep the terminal data automatically in synchronization with the community server's data by using a known, generic relational and transactional data synchronization mechanism. Also the distinction between Community Data and Identity Data is possible.

The communication between the client terminal and the servers is preferably at least in part wireless communication in order to provide a mobile terminal, but the communication can also be wireline communication. The "Service Assembly Point" (SAP) may be a server or a client terminal with a wireless or wireline connection to the Community Server.

The method according to the invention for managing data in a system comprising at least one community, at least one user, at least one community system comprising at least one database and at least one application, at least one service assembly point (SAP) comprising at least one database, and means for communication between a community system and a service assembly point, is characterized in that
- at least one service assembly point is a member of at least one community,
- the users can be members in a community with different profiles,
- and the users may use applications of a community according to said profiles.

The invention also relates to a storage media comprising a stored, readable computer program, which is characterized in that the program comprises instructions for controlling a data management system or components thereof to implement the method according to the invention.

The invention further relates to a data management system comprising at least one community and at least one user, at least one community system comprising at least one database and at least one application, at least one service assembly point (SAP) comprising at least one database, and means for communication between a community system and the service assembly point, which is characterized in that
- at least one service assembly point is a member of at least one community,
- the users of the SAPs can be members in a community with different profiles,
- and the users of the SAPs may use applications of a community according to said

profiles.

The invention further relates to a community server for a data management system, the community server comprising at least one community, at least one database, at least one application, and means for communication between the community system and a service assembly point (SAP) of the data management system, which is characterized in that the community server comprises

- means for joining service assembly points into communities,
- means for providing the users of the SAPs that are members in a community with different rights of use,
- means for allowing the users of the SAPs to use applications of a community according to said rights of use.

Some preferred embodiments of the invention are described in the dependent claims.

BRIEF DESCRIPTION OF THE DRAWINGS

Below the invention is described in more detail with reference to preferred embodiments shown as examples and to the enclosed figures, in which:

figure 1 shows some parts of an examplary system according to the invention;

figure 2 shows the basic units of an examplary system according to the invention;

figure 3 shows a flow diagram of examplary steps for joining a community and producing a set of applications to the client in a method according to the invention;

figure 4 shows a flow diagram of examplary steps for downloading and installing an application in a method according to the invention;

figure 5 shows a flow diagram of examplary steps for running an application in a method according to the invention; and

figure 6 shows two examplary publications in a system according to the invention.

DETAILED DESCRIPTION

Figure 1 shows an example of parts of a system 100 according to the invention.

The user has a locally runnable software and content of a smart terminal 120, which is in communication with the "Community Client", which is managed by one or multiple application and content management servers, "Community Server" 101, 107, and "Content Server", 103, 105, 109, respectively. The applications and pointers to web based applications are stored in the managed data storage, e.g. relational database, of the Community Servers. The content data of the applications may be managed and distributed using Content Servers. The identity data, such as User ID, name, access keys etc., of the user of the Community Client is maintained in the User Identity Client 130 of the Community Client terminal 120. The identity data may also be maintained in a separate Identity Server that is synchronized with the identity clients of the user (not shown in Fig. 1). The profile information of the users can be maintained in the Identity Server. The community replicas are stored in the databases 121, 123, 125, 127, 129 of the client terminal. The smart terminals can maintain a full or partial copy (replica) of the servers' data using suitable data synchronization technology, such as functionality disclosed in patent application document EP 0 860 788.

Figure 2 shows an examplary mobile Internet system according to the invention that consists of three main components: Community Server 201 for managing and classifying Services, Identity Server 202 for managing User Identity and Service Assembly Point (SAP) 220 that is typically a Smart Terminal but can also reside in any other type of network node. Such a network node may be e.g. a base station controller, access router, optical network router, etc. The invention provides a solution for the problem of bringing a user of the SAP a subset of locally executable or network based services that match the published user preferences and terminal capabilities of the currently used terminal.

Community Server

In the mobile Internet, there can potentially be a very large number of sources for the services. In this invention, these sources of services are called Mobile Communities. A service of the community can be network-based (i.e. it runs on a server that resides in the network) or it can be run locally in the Service Assembly Point.

The nature of the services can be described by classifying them using commonly agreed service classes. The service classes describe the nature of the service at different levels of detail. For instance, a commonly known "Tetris" game can belong to "TETRIS", "SPEED GAMES", "UNINTELLIGENT GAMES",

"GAMES" and "EVERYTHING" service classes.

In the invention, the services and their classifications are preferably managed by an entity called Community Server.

Identity Server

Each user of the Mobile Internet has his/her own identity. The Identity Data of the user can contain for instance following categories of data:

- Basic Identification Information has the unique identifier of the user, name and address information of the user, currently active role of the user etc. ;

- Service Preference Information contains the list of service classes that are of interest to the user;

- Access Keys to facilitate secure access to those services that require heightened security;

- Location and Service Usage History of the user, i.e. current and past locations where the user has been; and

- Terminals of the user, such as PCs, mobile phones, communicators & information appliances.

In the invention, the User Identity is preferably managed by an entity called Identity Server. The identity server can synchronize its data with the identity clients of one or multiple Service Assembly Points. Another possibility is to maintain the Identity Data only in the Service Assembly Point if there is no need to share identity information across devices.

Service Assembly Point

The Service Assembly Point is a node in the network, typically a Smart Terminal, where the list of services that is of interest to the user, which can be provided through the terminal that's currently in use, is stored.

When a user wants to assemble a service list from a community server to be used in a Service Assembly Point, following steps are taken:

1) To ensure that the Service Assembly Point has the most recent version of the

user identity data, it synchronizes its Identity Database Replica with the Identity Database Master that runs in the Identity Server. This step is not necessary if the SAP's version of the user's identity data is known to be the most recent version.

2) The Service Assembly Point establishes a connection with the Community Server's master database and creates a local replica of the database to the SAP. This step is not needed if the replica for that Community Server has been created earlier.

3) The Service Assembly Point publishes user's preference and terminal property data to the Community Server and invokes the service matchmaking process in the Community Server for instance by using data synchronization techniques such as SOLID Intelligent Transaction disclosed in document EP 0 860 788.

4) The service matchmaking process produces a list of services in the Community Server that is synchronized back to the community replica database of the Service Assembly Point.

5) The list of services is shown to the User.

6) To invoke a service, the User selects the service from the list.

7) If the service is to be executed locally in the Service Assembly Point, the service binaries and resources are downloaded from the Community Server to the local replica of the community using e.g. data synchronization techniques such as publications, unless downloaded already earlier. If the service is located by a network-based application server, the service is located and invoked by using the Uniform Resource Locator (URL) of the service.

Whenever the User needs to refresh the list of available services (e.g. when he/she changes preferences) or services of the community are changed, this can be done by re-executing steps 3-5 of the above sequence of steps.

Figure 3 shows a flow diagram of examplary steps for joining a community in a method according to the invention. The logical terminal 342 can establish a relationship with a new Community Server by registering itself with the server. At registration, 361, the database management system of the Community Client reserves a new area (e.g. database catalog) for the data of the new Community Server, 362. After creating the database catalog, the Community Client authenticates itself with the Community Server database, 363-367, using the locally maintained identity data and downloads meta-data about the Community Server's database to the new database catalog of the terminal. The meta-data contains information necessary to create the replica database schema, 365, and to synchronize the replica database later with the master database of the Community Server, 368.

Once the registration and meta-data download has successfully completed, the terminal database creates a database schema to the newly created catalog using scripts that have been sent from the community server to the terminal database as part of meta-data. Finally the preferences and terminal properties are matched with service classification and terminal requirements, 369, in a Service matchmaking process. The service matchmaking efficiently matches the user's preferences and available services of a community into a list of local and web-based services that is of interest to the user. After this, the new replica catalog can download community and application's header data from the community server's database by subscribing to Community publication.

Figure 4 shows a flow diagram of examplary steps for downloading and installing an application in a method according to the invention. The smart terminal can build its "desktop" i.e. links to its available services based on the data it has synchronized from the Community Server. This can be done for example by selecting a catalog from the Community Client's database, selecting a current role from the roles table of the catalog and listing the applications of the selected role in the user interface. The binary code, resources and installation scripts of the applications can be downloaded to the terminal, 470, separately by subscribing to a separate Application (APPLICATION_ID) publication, 471. In this publication, the APPLICATION_ID identifies the application whose binaries, resources and installation scripts are to be downloaded.

If the downloaded application requires a local Content database that is possibly synchronized with another database, the downloaded application's meta-data can contain scripts that create a separate database catalog, 472-474, for the content data of the application and register this new catalog with the master database of the Content Server. These scripts are executed, 475, after successful subscription of the APPLICATION publication.

Figure 5 shows a flow diagram of examplary steps for running an application in a method according to the invention. The user of the terminal can run an application by selecting the application from the user interface, 581. This invokes the application loader program that instantiates the selected application from the database tables to the main memory of the terminal, 582, and executes the program, 583, 584. Because the applications reside in the synchronizable local database, their consistency is always guaranteed by the generic data synchronization mechanism of the data management components of the Community Client and Server nodes. When application configuration is changed in the Community Server database, the new version is automatically downloaded to the Community Client when the databases are synchronized next time.

When the user does not need the services of the Community any more, the service suite of that Community can be deleted from the terminal simply by unregistering The replica database and by dropping the catalog and its content from the database.

Figure 6 shows a publication which can be used in implementing the present invention. Publication comprises a set of data in a database catalog that has been published in master database for synchronization to one or multiple replica databases. In the publication the USERS_APPS(user_ID) identifies the member user 692, the applications of the user 693 and the applications 694. In the publication, the APPLICATION(app_ID) 695 identifies the application 696 whose binaries 697, resources 698 and installation scripts 699 are to be downloaded to the client terminal.

A system according to the invention can be implemented by a person skilled in the art with state of the art information technology and communication technology components. A person skilled in the art can implement the functions according to the invention by arranging and programming such components to realize the inventive functions.

For example, it is preferable to implement the invention to work in a telecommunication system, which is complient with at least one of the following: TCP/IP, CDMA, GSM, GPRS, WCDMA, UMTS, Teldesic, Iridium, Inmarsat, WLAN and imode.

It is also preferable to use a standardized operating system in the terminals and servers. The operating system of a terminal can be, for example, Unix, MS-windows, EPOC, NT, MSCE, Linux, PalmOS and GEOS. The community server and/or the identity server may have at least one of the following operating systems: Unix, MS-windows, NT and Linux.

To a person skilled in the art it is obvious that in order to have an illustrative description the above presented exemplary embodiments have a structure and a function which are relatively simple. By applying the model presented in this application it is possible to design different and very complicated systems which, in obvious ways to the expert, utilise the inventive idea presented in this application.